Designing a Desktop Information System: Observations and Issues

Thomas Erickson and Gitta Salomon

Human Interface, Advanced Technology Group
Apple Computer, Inc.
20525 Mariani Ave., MS 76-3H
Cupertino, California 95014
(408)996-1010, thomas@apple.com, gitta@apple.com

ABSTRACT

This paper describes the first phase of a project to create a desktop information system for general users. The approach was to observe the problems, needs, and practices of several groups of information users, and to use these observations to drive the interface design of a prototype. In the first section of the paper, we describe problems which arise in the use of a relevance feedback system for information retrieval. In the second and third sections, we look at the needs and practices of users of both electronic and paper-based information systems. In the final section, we briefly describe the resulting design.

KEYWORDS: information retrieval, human interface, user interface, interactive systems, design process, design methodology, relevance feedback

INTRODUCTION

Today there are hundreds of on-line databases available to anyone with a personal computer and a modem. But it isn't very easy to access them. Each data source has its own interface; the computer often serves as only a terminal emulator. In most cases, while accessing information, users temporarily move into a world which is isolated from the rest of their computer environment. When they return, there are few facilities for working with the retrieved data.

In the future, users will want to move fluidly between numerous remote databases and effectively use the information they collect. Personal computers will need to be part of an integrated information environment.

In the Fall of 1989 we began a research project to explore interface issues related to the creation of just such an environment. Our focus was on problems that arise when general users are given access to a number of large, remote databases through their personal computers. (By "general user," we mean users who are not specialists in information retrieval; rather they need to obtain information to do their jobs.) One goal of the project, which is still underway, is the creation of a working prototype which will be installed in a real world environment, and the observation of its use. This prototype will give a group of accountants access to outside news sources and internal company data.

In this paper we discuss some of the interface issues which arose during the initial investigation phase and provide an illustration of how these issues drove an early prototype design. The investigation phase involved studying an existing commerical full-text information retrieval system, called DowQuest [5], which permits users to create powerful queries using natural language and relevance feedback [11] rather than a sophisticated query language. This phase also involved observation of information users. We interviewed and observed three groups of users: professional on-line searchers; day to day users of on-line information sources who were not information professionals; and a group of accountants. While the accountants made little or no use of on-line information sources, they nevertheless accessed and managed large amounts of paper-based information, and are the target group for the interactive prototype.

The remainder of this paper is divided into four sections. After a brief overview of the DowQuest system, we discuss issues concerning its query style. In the second and third sections, we look at the needs and practices of users of both electronic and paper-based information systems. Finally, we discuss a prototype that addresses some of these issues.

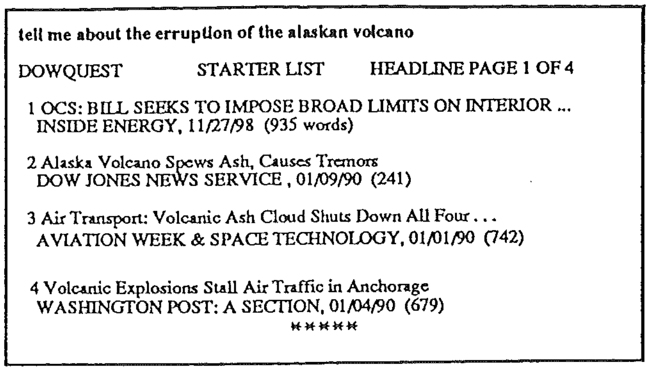
DOWQUEST AND RELEVANCE FEEDBACK

Early in the project, we were presented with the opportunity to use the DowQuest retrieval engine in our working prototype. In general, this engine seemed well suited to our target audience of accountants, who were generally lacking experience in the use of sophisticated query languages. Before we set out to design an interface to the engine, we examined the already functioning DowQuest implementation.

How DowQuest Works

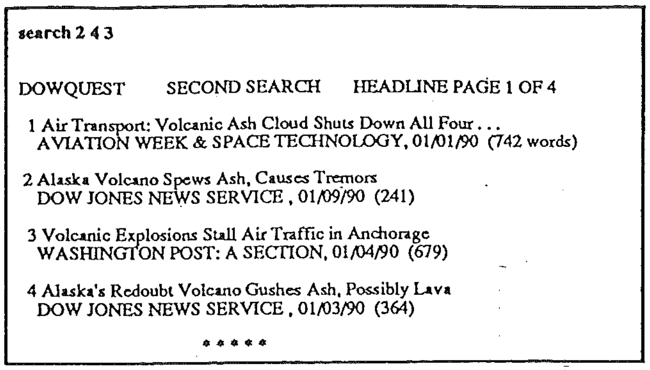
DowQuest, offered by Dow Jones & Company as part of their Dow Jones News Service, gives users access to over 350 news sources covering, approximately, the previous six months [5]. The system offers a full-text retrieval mechanism based on relevance feedback [12] which is purported to enable ordinary users to conduct powerful searches of large databases. Rather than using a sophisticated query language, DowQuest allows users to first type in a few words, get a list of potential hits, and then say – in essence – 'get more like that one.'

Figures 1 and 2 depict two phases of the process of constructing a query in DowQuest. In Figure 1, the user has entered a sentence describing the desired information.



```
tell me about the erruption of the alaskan volcano

DOWQUEST          STARTER LIST       HEADLINE PAGE 1 OF 4

 1 OCS: BILL SEEKS TO IMPOSE BROAD LIMITS ON INTERIOR ...
   INSIDE ENERGY, 11/27/98 (935 words)

 2 Alaska Volcano Spews Ash, Causes Tremors
   DOW JONES NEWS SERVICE , 01/09/90 (241)

 3 Air Transport: Volcanic Ash Cloud Shuts Down All Four . . .
   AVIATION WEEK & SPACE TECHNOLOGY, 01/01/90 (742)

 4 Volcanic Explosions Stall Air Traffic in Anchorage
   WASHINGTON POST: A SECTION, 01/04/90 (679)
                              *****
```

Figure 1. First phase of DowQuest interaction. *The user types in a 'natural language' query and the system searches the database using the non-'noise words' in the query. A title list for the 16 'most relevant' articles is returned.*



```
search 2 4 3

DOWQUEST       SECOND SEARCH      HEADLINE PAGE 1 OF 4

 1 Air Transport: Volcanic Ash Cloud Shuts Down All Four . . .
   AVIATION WEEK & SPACE TECHNOLOGY, 01/01/90 (742 words)

 2 Alaska Volcano Spews Ash, Causes Tremors
   DOW JONES NEWS SERVICE , 01/09/90 (241)

 3 Volcanic Explosions Stall Air Traffic in Anchorage
   WASHINGTON POST: A SECTION, 01/04/90 (679)

 4 Alaska's Redoubt Volcano Gushes Ash, Possibly Lava
   DOW JONES NEWS SERVICE , 01/03/90 (364)

                         *****
```

Figure 2. Second phase of DowQuest interaction. *The user instructs the database to find more articles like 2, 3 and 4, and the system returns a new set of relevant articles. Note that the first three, 'most relevant' articles are those that were fedback (an article is most 'like' itself); the fourth article is a new result.*

While DowQuest does not do actual natural language understanding, the user is encouraged to enter text in that manner. In the example shown, the system will drop out the words "tell," "me," "about," "the," and "of," and use the other, lower frequency words to search the database. After the user has entered the initial query, the system returns the titles of the 16 most 'relevant' articles, where 'relevant' is defined algorithmically and is based on a variety of features over which the user has no control (and often no knowledge). While this list frequently contains articles relevant to the user's query, it also usually contains items which appear – to the user – to be irrelevant. At this point, the user has the option of reading the articles retrieved or continuing to the second phase of the query process.

In the second phase of the process (Figure 2) the user tells the system which articles are relevant to the query. The user may either specify an entire article or particular paragraphs within it. The system takes the full text of the selections, drops out the noise words, and takes a limited number of the most 'informative' words for use in a revised query. It then returns a new list of sixteen relevant items. This second phase may be repeated as many times as the user wishes, though, in our observations, it was rare for users to iterate more than two or three times.

Interface Issues

Through observation of users, as well as our own experiences with the system, we uncovered a number of interface issues related to DowQuest's method of query specification and use of relevance feedback. A variety of lower level interface problems – such as the arbitrary 16 article result set size or the limitations of the teletype-style interaction – are discussed in [14]. We discuss two higher level problems which seem of general interest and importance.

Inappropriate Expectations of Intelligence

New users of DowQuest generally had high expectations of the system's intelligence. There are a variety of possible reasons for this, ranging from the seeming use of natural language, to the system's apparent ability to 'find more like this,' to the general belief in the intelligence of computers. In any event, these expectations were usually dashed when, in response to the first phase of the first query, DowQuest would return a set of articles containing many irrelevant articles. Consequently many users assumed the system was no good, or that no relevant articles existed, and would abandon the query before even trying relevance feedback [9].

Another negative effect due to the assumption of intelligence occured in the second phase of the query, when users requested the system to retrieve more articles 'like that one.' The new list of articles returned was ordered by 'relevance,' and, of course, no computer scientist would be surprised to find that an article is most similar to itself. General users, however, lacked this insight, and so when they looked at the new list and discovered that the first, most relevant article was the one they had told the system to find more like, they assumed there was nothing else relevant available and did not inspect the rest of the list [9]. While this assumption was incorrect, in human-human conversations it is conventional to assume that a provider of information will provide new information if it exists [8].

Ease of Use versus Control

Another problem, observed primarily in our own use of DowQuest, was one of undesired generalization. An example of this occurred for the query: 'tell me why Apple Computer stock prices have dropped.' The initial query produced some relevant articles, but after a couple rounds of feedback, the articles found veered away from Apple stock prices and began to emphasize the fluctuations in high technology stock prices. This occurred because articles discussing Apple's stock price tended to put it in a more general context, and repeated feedback of relevant articles reinforced this context. It is perhaps inaccurate to refer to such generalization as a problem, since it may often be a desired result. Nevertheless, it aptly illustrates the loss of control that results from shielding the user from the complexity of query languages.

While both problems discussed in this section arise in the context of DowQuest, analogs of them seem likely to occur in any system which attempts to use built-in intelligence to shield the user from underlying complexity.

NEEDS OF INFORMATION USERS

Through interviewing and observing users of both electronic and traditional information, we uncovered a number of issues that need to be addressed in the creation of an integrated desktop information environment. These are discussed n below.

The Need for Metaknowledge

Before users can create queries they need metaknowledge about the information in which they're interested. For example, they need to know 1) *where* to look for the answer to their question, and 2) *what* constitutes a reasonable question. This knowledge is not typically in the hands of the general user.

Choosing from 10,000 databases

There are many databases available on-line. How do users decide where to start looking for desired information? In observing expert on-line searchers at their weekly status meeting, we noted that a remarkable amount of time was spent sharing information about databases: topics included newly available databases, information quality, frequency of updates, timeliness of updates, costs, as well as situations in which a particular database should be consulted. Some of this information was gathered from experience, some gleaned from newsletters written by the database publishers. It became apparent that learning and memorizing database characteristics is a recognized part of the professional searcher's job.

Yet, a casual information user cannot be expected to stay abreast of database attributes in the same way. On the other hand, casual users often hold strong opinions about the quality of various data sources (whether well founded or not), and would likely be opposed to any system that automatically selected 'appropriate' databases. The information access system should, therefore, be designed to offer easy access to descriptive information about the available databases and offer aid in making decisions, when desired.

Asking a useful question

A related problem is that general users often lack familiarity with the amount or scope of knowledge associated with the information they are seeking. The on-line searchers indicated that it is not uncommon for a client to request, for example, all information about "artificial intelligence." In such situations, the searcher explains the difficulty and, through conversation, narrows the query's breadth. However, if the user addressed the same query to an on-line service, an enormous amount of material would be retrieved, unaccompanied by explanation. In such instances, the information system needs to help users make headway in their search. Various research systems have addressed this problem, and solutions range from providing the user with an example of a retrieved record to assist in query reformulation [15], to providing mechanisms for guiding the user through the information [10].

Additional information about these, and a variety of related issues, can be found in [2] and [3].

Working with Dynamic Information

Many databases contain frequently changing information. Bibliographic sources acquire new citations; news databases receive the latest reports. Over time, previously available information may longer be accessible. For example, due to the large volume of news items and storage limitations, DowQuest offers approximately the last six months of news at any one time. Several interface issues arise because of this dynamic nature of information sources, some of which are discussed in [1].

From our interviews we expect users will issue two types of queries: *ad hoc* queries, where they want an answer to a specific question and nothing more; and *on-going* queries, where they want to be kept up to date on a particular topic. The following examples illustrate problems that can occur in both of these cases.

One day in November of 1989, we issued the ad hoc query "earthquake volcano ashes seismic activity" on the DowQuest database. This query was successful and returned desired articles about the October 1989 California earthquake. However, when we executed the same query at a later date with the intent of quickly re-finding this information, we obtained articles about a newly erupting Alaskan volcano. Because DowQuest only returns 16 results to any query, the new information had taken precedence and the "California Earthquake" articles had slipped below the retrieval threshold. Even if DowQuest had displayed the entire result set, we may not have easily found the desired articles, because their location had changed. Users may find it disconcerting that on a different day the same query may not return the same set of results.

Similarly, a once useful on-going query may eventually become inadequate. For example, an on-going query established ten years ago to track news on portable computers might have performed well for quite some time. Today, the same query would return unmanageable numbers of articles. Furthermore, because terminology has changed, some relevant information might not be returned: machines that were called portable ten years ago might not be called portable today and many subclassifications now exist. In order to be useful again, the old query would have to be refined and narrowed to meet particular interests, in light of new developments. Possibly, several new, specific queries would be required to effectively deal with the information.

These problems are basically the result of a mismatch: a static query cannot remain effective when it is directed at a dynamic database. Therefore, the query interface will need to establish a means of explaining why and how changes have occurred and offer ways for the user to easily alter the query as the available information changes.

PRACTICES OF INFORMATION USERS

In our observations of general information users, we noted a number of practices which seemed of importance in their use of information. It seems likely that any successful desktop information system will have to support such practices.

Skimming

In our study of accountants, we found that whether they were dealing with newspapers, technical papers, or memos, no one ever used the verb "read." These users began by skimming all information they received, often relying on the layout of the information to give them a quick overview. Only rarely did they decide to read the material thoroughly. One accountant subscribed to approximately 20 magazines and journals, but infrequently ventured beyond the table of contents. Similar usage patterns have been noted in other domains [4].

It is difficult to skim electronically-based information in the same way. One accountant, who had personally implemented part of an electronic database of a standard accounting reference, confessed that he preferred using the hard copy version because it was easier to skim.

One way to facilitate skimming is to provide article summaries. However, it is often not possible to summarize (either automatically or manually) a document because different people will look for different types of information. The accountants we interviewed noted that they often search for information that is implicit or even deliberately concealed (such as bad financial indicators), and would be even less likely to be included in an abstract.

A different tactic is to rely on structure in the document itself. Various designers (e.g., [7]) have argued that document usability can be enhanced by incorporating the structure of traditional documents into on-line information. Paper-based documents such as magazines employ a variety of visual design techniques which could be used to facilitate skimming in on-line documents. The design challenge here is to support skimming in ways that go beyond adaptation of traditional printed media design and take advantage of the properties of electronic media (e.g., [6]). For example, one accountant suggested that the system could display the first few sentences of every paragraph and he could choose where to expand to full text.

Annotation

Most of the accountants annotated (i.e., added comments or marked-up) the paper-based information they saved. Annotation was used as a memory cue about what aspects of the information were of importance. In addition, annotation was used to add value. For example, annotation facilitated skimming by other people with whom the document was shared. Also, it was used to indicate relationships between the document and other information.

Currently, it's difficult to annotate an electronic document casually. One accountant who maintained information on-line went to great lengths to annotate it. He would import the ASCII text into a word processor and mark it up by changing text styles to bold or underline. More typically, users printed the information they'd found, marked it up by hand, and filed it, thus losing any capacity for electronically managing the retrieved documents. A complete information environment needs to provide users with annotation tools, the means to view documents in both pristine and annotated form, and the ability search for elements in both the original data and the annotations.

Our interviews with accountants also revealed a way in which annotation may be more important in an electronic environment than in a paper-based one. The accountants themselves are audited by corporate level quality control people who want to make sure that they're performing to the company's standards. Among other things, quality control people look at clipping files to ensure that the accountant is keeping up on the industry and clients. Future systems which automatically retrieve information on particular topics would eliminate this as a source of evidence. In such an instance, the existence of annotations would provide proof that the information had been 'touched by human hands'– evidence that might be welcomed by clients as well as quality controllers.

Organization

The accountants discarded all but the most important information; space constraints, as well as the difficulty of deciding which file folder was most appropriate, deterred them from saving more. There was a general feeling that the fewer items saved, the easier it was to re-locate them. One of the few users who maintained information in electronic form saved items into a "scrapbook" file, but rarely revisited anything because this required a sequential scan through the file. These cases indicate that an information management system needs to supply users with tools to organize and reorganize their data, once retrieved.

Such tools need to support full text search on saved items, as well as the ability to search on other criteria. For example, users often remember the approximate date on which the data was found, or the source it came from. Tools provided by the system should allow the use of combinations of such attributes for searching and reorganizing, thus permitting users to create their own idiosyncratic databases with items retrieved from external databases.

FROM OBSERVATIONS TO DESIGN

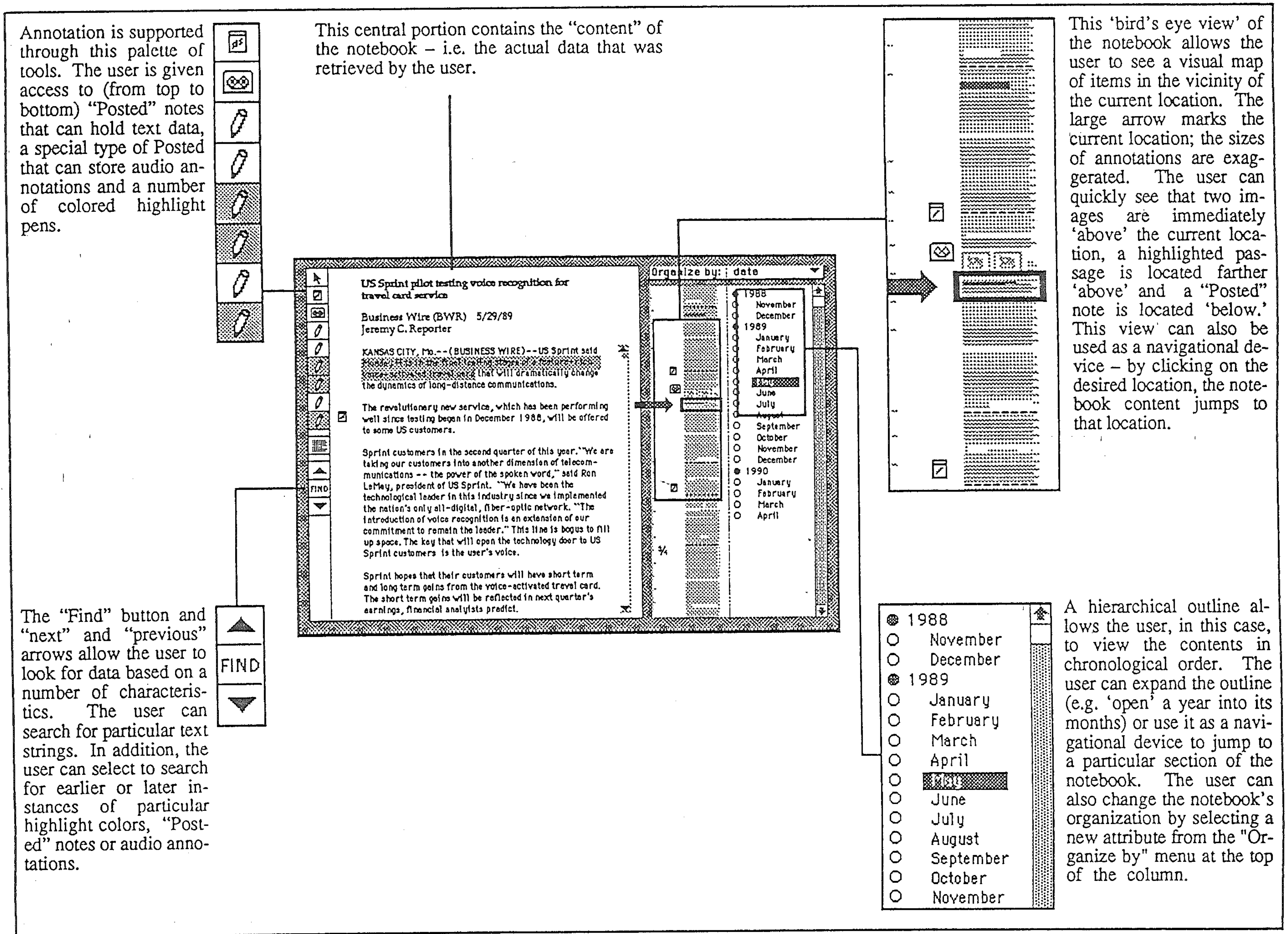
In this section, we briefly describe some of the design elements which resulted from consideration of the issues previously identified. Note that the design does not address all of the issues we have discussed in this paper. Furthermore, we must emphasize that – because the system is still being implemented and has yet to be tested on the intended users – we cannot say whether the features we describe will be successful. Readers may wish to look at related systems, such as SuperBook [6] and Concordia [13], which have already progressed through implementation and testing phases and which address similar issues.

The Prototype

Our prototype interface design has three components: *reporters*, *newspapers*, and *notebooks*.

Reporters are what users interact with to define the type of information they wish to retrieve. Through a form-based dialogue, a user can give a reporter specifications, examine items it retrieves, and use relevance feedback to refine those specifications. Any reporter can be automated so that it will access desired databases on a regular basis.

By using a reporter metaphor, we hope to provide users with a way to understand and contend with a less-than-predictable query mechanism and the dynamic nature of databases. This metaphor allows us to examine an interesting conjecture: anthropomorphism may be useful for representing ignorance, as well as intelligence. Users were often disturbed when initial queries to DowQuest would result in the retrieval of irrelevant articles, and sometimes concluded that "the system" didn't work. Would they be more forgiving of a reporter and expect it to improve with feedback? In addition, real-world reporters embody many of the characteristics of the retrieval mechanism: the ability to use fuzzy information as feedback ('find more like that one'), and the ability to function in a world of changing information (a reporter is not expected to come back with the same information next week).



Annotation is supported through this palette of tools. The user is given access to (from top to bottom) "Posted" notes that can hold text data, a special type of Posted that can store audio annotations and a number of colored highlight pens.

This central portion contains the "content" of the notebook – i.e. the actual data that was retrieved by the user.

This 'bird's eye view' of the notebook allows the user to see a visual map of items in the vicinity of the current location. The large arrow marks the current location; the sizes of annotations are exaggerated. The user can quickly see that two images are immediately 'above' the current location, a highlighted passage is located farther 'above' and a "Posted" note is located 'below.' This view can also be used as a navigational device – by clicking on the desired location, the notebook content jumps to that location.

US Sprint pilot testing voice recognition for travel card service

Business Wire (BWR) 5/29/89
Jeremy C. Reporter

KANSAS CITY, Mo.--(BUSINESS WIRE)--US Sprint said [illegible] that will dramatically change the dynamics of long-distance communications.

The revolutionary new service, which has been performing well since testing began in December 1988, will be offered to some US customers.

Sprint customers in the second quarter of this year. "We are taking our customers into another dimension of telecommunications -- the power of the spoken word," said Ron LeMay, president of US Sprint. "We have been the technological leader in this industry since we implemented the nation's only all-digital, fiber-optic network. "The introduction of voice recognition is an extension of our commitment to remain the leader." This line is bogus to fill up space. The key that will open the technology door to US Sprint customers is the user's voice.

Sprint hopes that their customers will have short term and long term gains from the voice-activated travel card. The short term gains will be reflected in next quarter's earnings, financial analysts predict.

The "Find" button and "next" and "previous" arrows allow the user to look for data based on a number of characteristics. The user can search for particular text strings. In addition, the user can select to search for earlier or later instances of particular highlight colors, "Posted" notes or audio annotations.

A hierarchical outline allows the user, in this case, to view the contents in chronological order. The user can expand the outline (e.g. 'open' a year into its months) or use it as a navigational device to jump to a particular section of the notebook. The user can also change the notebook's organization by selecting a new attribute from the "Organize by" menu at the top of the column.

Figure 3. Prototype design for information "notebook." *This screen depicts a notebook in which a user can skim, search, organize and annotate information.*